

context instance is created upon successful identification of the user. The user context instance also includes information about the user including the roles and a unique user identifier. A request is received from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component such that the user context instance is passed as a parameter from the first component to the second component, and wherein completion of the second service is necessary to complete the first service. The user context is queried for ~~the information about the~~ a unique user identifier. ~~The user information-unique user identifier~~ is compared with an access control list for verifying that the user has access to the first component. The user information-unique user identifier is also compared with an access control list for verifying that the user has access to the additional-second service of the other-second component.

Page 4, lines 16-19

In one embodiment of the present invention, all user interactions may be logged as well. In another embodiment of the present invention, a user interface may be modified to provide access to actions that can be performed by the user based on the unique user identifier ~~an identity of the user~~ and the roles associated with the user.

Page 4, lines 21-25

In even a further embodiment of the present invention, the user context instance may also be passed along as a parameter of service invocations. Additionally, the first service invoked may associate any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance. In another aspect of the present invention, the user context instance may also encapsulate security certificates of the user.

Amendments to the Abstract

A system[[,]] and method ~~and article of manufacture~~ are provided for maintaining a security profile throughout nested service invocations on a distributed components component-based system. Interconnections are provided between distributed components each having nested service invocations. ~~A user is identified. The user is associated with roles.~~ A user context instance is created upon successful identification of the user. ~~The user context instance~~